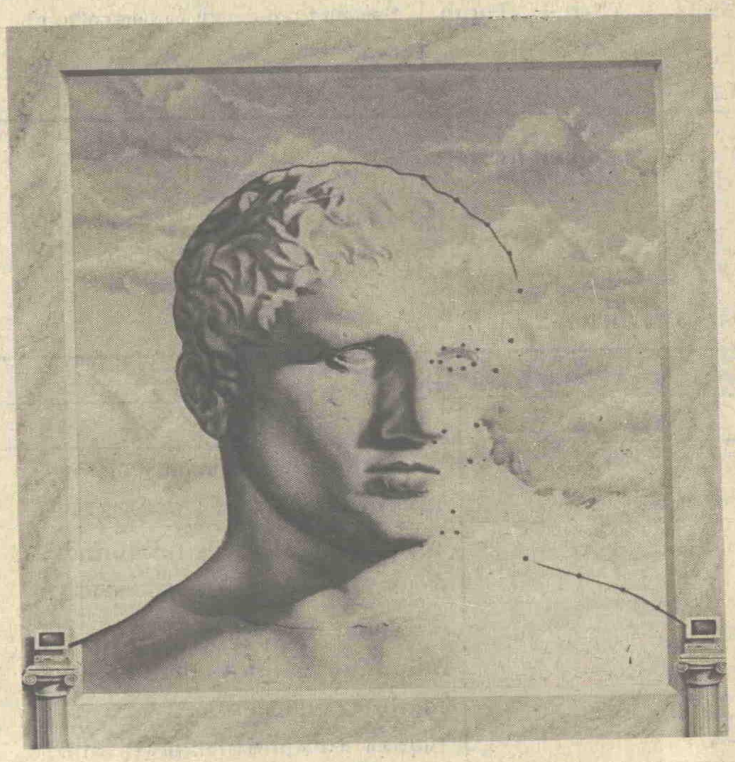
ซอฟต์แวร์
ลออ เพิ่มสมบัติ

เริ่มต้นการเขียนโปรแกรม กราฟิก อย่างมีระบบ ตอนที่ 4 : การเขียนกราฟิกโดยอาศัยรอมไบออสด้วยเทอร์โบซี

เทคนิคของภาษาซี ที่โปรแกรมเมอร์ ใช้สร้างภาพบนคอมพิวเตอร์ ที่ปฏิบัติการบนดอส



ความสามารถในการวาดรูปลงบนจอสีใช้เวลาเพียงเสี้ยววินาที เป็นความสามารถของคอมพิวเตอร์ยุคใหม่ รูปกราฟจะช่วยให้ผู้ใช้รายงานผลจากการเขียนโปรแกรมฐานข้อมูล หรือจากแผ่นตารางทำการ เกมทางคอมพิวเตอร์ วินโดว์

ในตอนนี้เราจะกล่าวถึงเทคนิคของภาษาซีที่โปรแกรมเมอร์ใช้ในการสร้างจุดภาพบนคอมพิวเตอร์ที่ปฏิบัติการบนดอส ด้วยการเข้าถึงตรงโดยใช้รูทีนรอมไบออสและวิดีโอเมมโมรีซึ่งเป็นรูทีนที่เร็วที่สุดในการแสดงผลทางกราฟิก ดังนั้นคุณควรมีตัวปรับภาพมาตรฐานอันได้แก่ ซีจีเอ, อีจีเอ หรือวีจีเอ อย่างใดอย่างหนึ่ง

โหมด (Modes)

นักศิลปควรมีโอกาสเลือกสื่อเพื่อจะสร้างภาพได้หลายสื่อ (สีน้ำมัน สีน้ำ เครื่องมือแกะสลัก และอื่น ๆ) นักเขียนโปรแกรมคอมพิวเตอร์ทางด้านกราฟิกก็เช่นเดียวกัน ควรจะเลือกโหมดหรือรูปแบบวิธีได้ต่าง ๆ กัน ซึ่งแต่ละโหมดจะมีวิธีเลือกรูปลักษณะของกราฟิกต่าง ๆ กันออกไป ลักษณะที่ต่างกันนี้จะรวมถึงความคมชัด จำนวนของสี การรายงานผลรูป หรือข้อความ โหมดที่ต่างกันต้องการฮาร์ดแวร์ที่ต่างกัน (อาทิ จอภาพและตัวปรับต่าง ๆ) และยังต้องมีวิธีการเข้าสู่การเขียนโปรแกรมที่ต่างกันออกไปอีก ทั้งหมดนี้มักจะก่อให้เกิดความสับสนแก่โปรแกรมเมอร์กราฟิกมือใหม่ ดังนั้นจึงควรจะถกกันก่อนเกี่ยวกับธาตุแท้ของกราฟิก และวิธีการรวมโหมดเข้าด้วยกัน

ความคมชัด (Resolution)

ภาพบนจอคอมพิวเตอร์ประกอบด้วยจุดเล็ก ๆ ที่เรียกว่าจุดภาพ (pixels ย่อมาจาก picture elements) จุดภาพจัดวางเป็นแถวแนวนอนบนจอและมีจำนวนแถวจำกัด แต่ละแถวประกอบ

ด้วยจำนวนจุดภาพที่แน่นอน จำนวนจุดภาพที่ใช้บนจอ จะเรียกว่าความคมชัด (Resolution) ซึ่งแต่ละโหมดของกราฟิกจะใช้ความคมชัดต่างกัน เช่น โหมด 4 ใช้ความคมชัดแบบ 200 แถว และแต่ละแถวมี 320 จุดภาพ เรียกสั้น ๆ ว่า 320 × 200

ความคมชัดสูง หมายความว่าภาพคม ภาพชัด มีความเป็นขั้นบันไดน้อย ถ้าลากเส้นตรงตามเส้นทแยงมุมบนจอภาพ (jaggies)

ข้อความหรือรูปกราฟิก

บางโหมดจะเอาข้อความไว้บนจอสีได้ แต่บางโหมดก็ใช้ได้กับกราฟิกเท่านั้น ในโหมดกราฟิก แต่ละบิตหรือแต่ละกลุ่มของบิตจะสอดคล้องกับจุดภาพเฉพาะบนจอภาพ

สี

บางโหมดกราฟิกจะจัดสีให้มาก ๆ กว่าโหมดอื่น มีหลายโหมดมีสีเพียงสองสี คือสีดำกับสีขาว (หรือบางทีก็เป็นสีดำ สีเขียว หรือ อำพัน ขึ้นอยู่กับคอมพิวเตอร์ที่คุณมี) โหมดอื่นอาจจะมี 4 หรือ 16 หรือ 256 สี

แต่บางสถานการณ์มักจะซับซ้อนมากกว่านี้ เช่น โหมดเฉพาะที่อนุญาตให้ใช้เพียง 4 สีบนจอ แต่อาจจะสลับเป็นกลุ่มที่สองของสีทั้งสี่ได้ (ถึงแม้ว่าสีกลุ่มที่สองจะไม่มีอยู่บนจอภาพขณะนั้นก็ตาม) ซึ่งกลุ่มของสีเหล่านี้เรียกว่า palettes

จอแสดงภาพ

จอที่ต่างกันจะใช้กับโหมดการแสดงผลที่ต่างกัน จอที่ธรรมดาและถูกที่สุดคือจอที่แสดงข้อความเท่านั้น ซึ่งจะใช้ได้กับโหมด 7

โหมดซีจีเอใช้ได้กับ 0 ถึง 6

โหมดอีจีเอจะเพิ่มโหมด 13 ถึง 16

โหมดวีจีเอจะเพิ่มโหมด 17 ถึง 19 ถ้าโหมดยิ่งสูงความคมชัดก็จะมากขึ้น

ตัวปรับการแสดงผล

โดยทั่วไป (ยกเว้นชุด IBM PS/2 บางชุด) คอมพิวเตอร์พีซีไม่มีสมรรถนะที่จะส่งสัญญาณที่จำเป็นในการทำให้เกิดสภาวะข้อความ หรือสภาวะกราฟิกบนจอภาพได้ จำเป็นต้องมีตัวปรับการแสดงผล ซึ่งจะซื้อมาใส่ไว้ในช่อง (slot) ที่อยู่ในเครื่อง โดยมีอยู่ 4 แบบมาตรฐาน ได้แก่ จอโมโนโครม ซีจีเอ, อีจีเอ และวีจีเอ

ซีจีเอให้ความคมชัดเป็น 640×200 ที่ความคมชัดระดับนี้จะมีสองสีที่ใช้ได้ (ดำ และขาว) 300×200 ใช้ 4 สีได้

ซีจีเอมีโหมดข้อความหลายโหมด

อีจีเอเพิ่มความคมชัดเป็น 640×350 ซึ่งใช้สีได้ 16 สี

วีจีเอเพิ่มความคมชัดเป็น 640×480 ซึ่งใช้สีได้ 16 สี

ถ้าจะใช้ 256 สี จะต้องลดความคมชัดลงเป็น 320×200

Display memory การแสดงความจำ

ตัวปรับแสดงผลที่ต่างกันจะใช้ความจำมากน้อยต่างกันด้วย

โมโนโครมใช้ความจำ 4 k ซึ่งก็เพียงพอที่จะใช้กับ 2 ไบต์ สำหรับแต่ละ 2000 อักษร (80 คอลัมน์คูณกับ 25 แถว)

ซีจีเอมีความจำมากกว่า คือ 16 k ก็เพียงพอสำหรับ 640×200 จุดภาพ ถ้าใช้เพียง 2 สี ในกรณีแต่ละบิตในความจำจะสอดคล้องกับ 1 จุดภาพ รูปภาพ 16 k จะรับมาจากจำนวนจุดภาพทั้งหมด คือ 640×200 ซึ่งเท่ากับ 128,000 จุดภาพ เนื่องจากแต่ละจุดภาพสอดคล้องกับ 1 บิต ลองหารด้วย 8 ดู (8 บิตเท่ากับ 1 ไบต์) จะได้ 16000 ไบต์ (ซึ่งมีค่าต่ำกว่า 16 K = 16,384 ไบต์) ถ้าตัดความคมชัดลงครึ่งหนึ่งเป็น 320×200 เราก็ใช้ 2 บิต ต่อ 1 จุดภาพได้ และใช้ 2 บิต แทนสี่สี ก็จะเป็นอีกโหมดหนึ่งที่ใช้ได้กับซีจีเอ ในทางตรงกันข้ามวีจีเอมีรูปแบบมาตรฐานของความจำเป็น 64 k ถ้าใช้ 16 สีกับความคมชัด 640×200 (การพิสูจน์ก็ทำแบบข้างต้นแต่อย่าลืมว่า 16 สีต้องใช้ 4 บิตต่อ 1 จุดภาพ) ถ้าจะ

เสริมความจำให้แก่อีจีเอก็ทำได้เป็น 128 k หรือ 256 k ถ้าเป็น 128 k โหมดอื่นก็น่าจะเป็นไปได้ โดยมีความคมชัด 640×350 พร้อมสี 16 สี (การขยายความจำเป็น 256 k ทำได้โดยการเพิ่มเพจ (Pages) เข้าไปอีก จะได้อธิบายละเอียดต่อไป)

ส่วนวีจีเอจะมีความจำ 256 k ความคมชัด 640×480 และใช้สี 16 สี 320×200 กับสี 256 สี

ตารางที่ 1 สรุปเกี่ยวกับความจำที่จำเป็นของจำนวนสี และความคมชัดต่าง ๆ กัน

แอดเดรส

เหมือนกับส่วนอื่น ๆ ของแรม ตำแหน่งที่อยู่ในความจำของตัวปรับกราฟิก เรียกว่าแอดเดรสเช่นกัน โดยตำแหน่งแรกของระบบความจำในโมโนโครมจะเริ่มที่แอดเดรส B0000

–ซีจีเอเริ่มที่ B8000
–อีจีเอและวีจีเอเริ่มที่ A0000

แอดเดรสข้างต้นสำคัญมากถ้าเราต้องการเข้าถึงความจำแสดงผลโดยตรงจากโปรแกรม

ตารางที่ 1 ตารางสรุปเกี่ยวกับความจำที่จำเป็นของจำนวนสีและความคมชัดต่าง ๆ

	จำนวนสี		
ความคมชัด	2	4	16
320 × 200	8 k	16 k	32 k
640 × 200	16 k	32 k	64 k
640 × 350	32 k	64 k	128 k

เพจ (Pages)

เมื่อความจำทางด้านกราฟิกที่มีอยู่ในตัวปรับเพียงพอก็สามารถจะรักษาจอภาพของข้อมูลไว้หลายจอภาพในความจำ ณ เวลาเดียวกันนั่นคือ บนบอร์ดอีจีเอโหมด 13 (320×200, 16 สี) ต้องใช้ความจำ 32 k แต่จะใช้ 64 k ก็ได้ ซึ่งถือว่าเป็นความจำพิเศษที่ถูกใช้เก็บจุดภาพบนจอครั้งที่สอง โดยการสลับหน้าหนึ่งไปอีกหน้าหนึ่งอย่างรวดเร็วก็จะทำให้จุดภาพเปลี่ยนแปลงบนจอได้

การใช้โหมด

ในบทความนี้จะเน้นถึงโหมด 4 ซึ่งใช้ได้กับตัวปรับกราฟิกทุกตัวปรับ ส่วนโหมดอื่น อาทิเช่นโหมด 13 จะใช้กับตัวปรับอีจีเอและวีจีเอ โหมด 18 และ 19 จะใช้ได้เป็นพิเศษกับวีจีเอเฉพาะ ดังนั้นเราจึงเริ่มดูที่โหมด 4 ก่อน แต่ก่อนที่จะลงมือดูควรจะรู้ถึงวิธีสลับจากโหมดหนึ่งไปยังอีกโหมดหนึ่ง จึงจะเรียนรู้การใช้โหมดได้ดีขึ้น

การตั้งโหมด

ถ้าคุณมีจอภาพโมโนโครมและตัวปรับโมโนโครม เมื่อเริ่มเปิดเครื่องจะบูตที่โหมด 7 ถ้าคุณใช้ซีจีเอ อีจีเอ หรือวีจีเอ จะบูตที่โหมด 3 (ข้อความอย่างเดียว) ซึ่งโหมดที่กล่าวมาแล้วจะไม่อนุญาตให้คุณเล่นกราฟิก

ถ้าจะวาดกราฟิกบนจอสีจะต้องรู้วิธีเปลี่ยนโหมด 7 (หรือโหมด 3) ให้เป็นโหมดกราฟิก

คำสั่งการตั้งโหมดในรอมไบออส (ROM BIOS)

การเลือกโหมดเราจะใช้รูทีนของรอมไบออสเรียกโหมด การตั้งวิดีโอซึ่งก็จะต้องเซต 0 ให้แก่รีจิสเตอร์ AH ตัวเลขโหมดที่ต้องการให้แก่รีจิสเตอร์ AL พร้อมกับเริ่มต้นทำงานกับการขัดจังหวะให้เป็น 10 (เลขฐาน 16)

รูทีนในรอมไบออสตั้งโหมดวิดีโอ

อินเตอร์รัพต์ 10 (ฐาน 16) ; วิดีโอ

Input Registers : AH = 0

AL = mode number

Output Registers: NONE

ดูที่โปรแกรม SETMODE.c ในรูปที่ 1

โปรแกรมนี้ช่วยในการเปลี่ยนโหมดเข้าสู่กราฟิก เช่นต้องการให้เป็นโหมด 4 ที่ดอสก็เขียนคำสั่งดังนี้

C > Setmode 4

(อย่าใช้โปรแกรมนี้ขณะที่อยู่ใน IDE ของเทอร์โบซี ถึงแม้คุณจะรันได้จากดอสเชลล์ก็ตาม ควรจะออกมาจาก IDE ก่อนจึงจะใช้)

เมื่อจะพิมพ์ C> Setmode จะโปรแกรมบนจอสี แต่คุณจะไม่เห็นเคอร์เซอร์ เนื่องจากโหมดกราฟิกไม่ได้สนับสนุนเคอร์เซอร์ ถ้าต้องการเห็นเคอร์เซอร์ก็ต้องกลับไปที่โหมดข้อความก่อโหมด 3

การแสดงจุดจอภาพด้วยรูทีนรอมไบออส

คุณคงรู้จักกราฟิกดีขึ้นใช่ไหม และวิธีเปลี่ยนโหมดอีกด้วย ต่อไปนี้เราจะลองมาใส่จุดภาพลงจอกันดีกว่าวิธีง่ายที่สุดในการแสดงภาพก็คือใช้รอม-รูทีนที่มีชื่อว่า write Dot และรูทีนรอมสำหรับการแสดงไม่ได้ทำให้เร็วขึ้นมากนัก แต่ทำให้ง่ายต่อการเขียนโปรแกรม โปรแกรมต่อไปนี้ estripes.c (ดูรูป 2) จะถูกออกแบบให้ใช้ได้กับโหมด 4 (โหมดนี้สามารถจะใช้ได้กับทุกตัวปรับ) แต่การที่จะให้โปรแกรมทำงานจะต้องใช้ Setmode.c ที่เราได้เขียนขึ้นข้างต้นเปลี่ยนเป็นโหมด 4 ก่อน

โปรแกรม estripes.c จะวาดแถบแถบตามแนวตั้งบนจอด้วยสีต่าง ๆ น 4 สี (ดูรูปที่ 2)

```
/*setmode.c*/
/*sets graphics mode to value supplied*/
#include<dos.h>
#define SETMODE 0 /* "set video mode" servive */
#define VIDEO 0x10  /*video BIOS interrupt number*/

void main(int argc,char *argv[])
{
union REGS regs ;
int mode;

if (argc !=2)
 { printf("Example usage:setmode 7"); exit();}

 mode=atoi(argv[1]);   /*string to integer*/
 regs.h.al=(char)mode; /*mode number in AL register*/
 regs.h.ah=SETMODE;    /*service # in AH register*/
 int86(VIDEO,&regs,&regs); /*call video interrupt*/
 }
```

รูปที่ 1
โปรแกรม Setmode.c

```
/* estripes.c*/
/*files EGA screen with 16 colors bars .Use mode 13 (320x200)*/
#include <dos.h>
#define MAXR 200
#define MAXC 320
#define VIDEO 0x10
#define WDOT 12
void main(void)
{
 union REGS regs;
 int row,col;
 for(row=0;row<MAXR;row++)
  for(col=0;col<MAXC;col++)
   {
   regs.h.ah=WDOT;
   regs.x.dx=row;
   regs.x.cx= col;
   regs.h.al=col/20;
   regs.h.bh=0;
   int86(VIDEO,&regs,&regs);
  }
 }
```

รูปที่ 2
โปรแกรม estripes.c
จะวาดแถบสี 4 แบบ

```
Rom Bios routine ; write dot
Interrupt 10 Hex : video
Input registers   : AH  = oc hex
                    CX  = Column number
                    DX  = Row number
                    AL  = color
                    BH  = page number
output registers  : NONE
```

รูปที่ 3

ตารางที่ 2 ค่าทั่วไปในการเซตสี

หมายเลข	สี
0	black
1	cyan
2	magenta
3	white

ค่าทั่วไปหรือค่าโดยปริยายในการเซตสีในโหมดนี้คือ

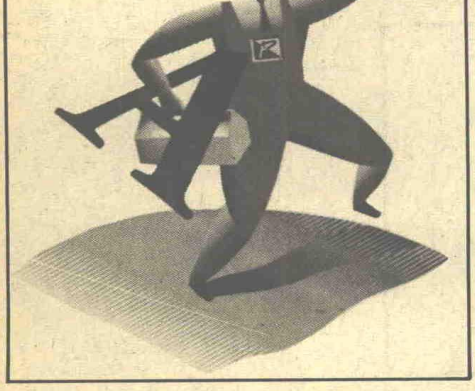
ดูในตารางที่ 2

ถ้าจะบ่งชี้สีสำหรับจุดภาพโดยเฉพาะคงจะต้องใช้เลขที่บอกถึงเฉพาะสีนั้น ๆ จาก 0 ถึง 3 ในรีจีสเตอร์ AL เมื่อเรียกใช้รูทีน Write Dot Rom Bios ในโปรแกรม estripes.c จะสร้างแถบตามแนวตั้ง 4 แถบ ซึ่งแต่ละแถบจะมีสีต่างกัน

จำนวนคอลัมน์ (ตัวแปร col) จะมีช่วง 0-319 ดังนั้นเราจึงต้องหารด้วย 80 ซึ่งจะทำให้สีอยู่ในช่วง 0-13 โดยใช้ประโยคคำสั่ง

```
regs.h.al = col/80;
```

ทำให้แถบ 4 แถบถูกวาดขึ้นบนจอ เนื่องจากแถบแรกเป็นสีดำที่อยู่ซ้ายสุดบนจอภาพ



```
/*conrect.c*/
/*draw concentric rectangle of different colors*/
#include <dos.h>
void rect(int,int,int,int, unsigned char);/*prototype*/
void putpt(int,int,unsigned char);

void main(void)
{
 int z;
 for(z=50;z>=10;z-=10)   /*series of rectanges*/
  rect(100-z,100+z,160-z,160+z,(z/10)%4);
 }
 /*rect()*/
 /*draws rectange on the screen using putpt()*/
 void rect(int top,int bot,int left,int rite,unsigned char color)
 {
 int x,y;
 for(y=top;y<=bot;y++)
  for(x=left;x<=rite;x++)
   putpt(x,y,color);
 }
 /*putpt()*/
 /*displays point at location col,row*/
 #define VIDEO 0x10
 #define WDOT 0x0c
 void putpt(int col,int row,unsigned char color)
 {
 union REGS regs;

 regs.h.ah=WDOT;
 regs.x.dx=row;
 regs.x.cx=col;
 regs.h.al=color;
 regs.h.bh=0;
 int86(VIDEO,&regs,&regs);
 }
```

รูปที่ 4
โปรแกรม conrect.c

เราทราบมาแล้วว่าการใช้ Write Dot Rom Bios นี้ไม่เร็วนัก แต่จะขึ้นอยู่กับตระกูลของ PC ที่คุณใช้ เครื่องที่ช้าที่สุดคือ PC XT ที่รันภายใต้ 4.77 เมกะเฮิรตซ์ ใช้เวลา 40 วินาที ในการแสดงภาพจากโปรแกรม estripes.c ถึงแม้ว่าจะเป็นเครื่อง 386 ก็ใช้เวลาหลายวินาทีเหมือนกันในการแสดงผลซึ่งเราจะทำให้เร็วขึ้นทีหลัง

การวาดรูปสี่เหลี่ยม

ตอนนี้เรารู้แล้วว่าจะลงจุดบนจอภาพอย่างไร เราอาจจะสร้างรูปที่ไม่ใช่แถบลงมาตามแนวดิ่ง แต่อาจจะเป็นรูปสี่เหลี่ยมซ้อน ๆ กันบนจอภาพ ตา โปรแกรม conrect.c (ดูรูปที่ 4) ซึ่ อาจจะแบ่งโปรแกรมออกเป็น 3 ส่ว คือ ฟังก์ชัน putpt( ) ที่ใช้รูทีนรอม ไบออสลงจุดบนจอภาพ ฟังก์ชัน re ( ) รูปสี่เหลี่ยม

ฟังก์ชัน main ( ) จะเรียกใ rect ( ) ห้าครั้งด้วยกัน

การให้สี Palette กับสีพื้น

เรารู้อยู่แล้วว่าเรามีวิธีการเลือก ได้ 4 สีตามใจปรารถนา วิธีการเลือกนี้ เป็นอย่างไรลองมาดูกัน

รูปที่ 5
นโหมด 4 จะมีวิธีเลือกสีได้ 2 สี

```
Rom bios routine : set color palette
Interrupt 10 hex : Vedio
Input registers : AH  = 0B (hex)
                  BH  = 1 (เปลี่ยน palette)
                  BL  = Palette number
Output register : NONE
```

ในโหมด 4 จะมีวิธีเลือกสีได้ 2 สี หรือ palettes (ดูรูปที่ 5)

โปรแกรมที่อนุญาตให้เปลี่ยน palette ตามต้องการคือโปรแกรม setpal.c (ดูรูปที่ 6) ซึ่งผลลัพธ์ที่ได้จะเป็นรูปสี่เหลี่ยมผืนผ้าบนจอภาพใน palette โดยปริยาย คือใช้เลข 1 แล้วสับเปลี่ยนไปใช้ palette ที่สอง

รูปที่ 6
โปรแกรม setpal.c

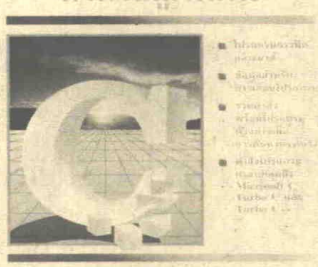
```
/*setpal.c*/
/*sets color palette to one of two values*/
#include <dos.h>      /*declares REGS*/
#define SETPAL 0x0B   /*"set color palette" service*/
#define VIDEO 0x10    /*video BIOS interrupt number */
void main(int argc,char *argv[])
{
 union REGS regs;
 int pal;          /*palette number*/

 if(argc !=2)
  {printf("Example usage:c>setpal 0"); exit(0);}
 pal =atoi(argv[1]);     /*string to integer*/
 regs.h.bh=1;            /*BH=1 to set palette*/
 regs.h.bl=pal;          /*palette number from user*/
 regs.h.ah=SETPAL;       /*service # in AH register*/
 int86(VIDEO,&regs,&regs);/*call video interrupt */
}
```

```
c> setmode 4
c> conrect
c> setpal 0
```

ซีเอ็ด

วางตลาดแล้ว

หนังสือเล่มนี้ได้กล่าวถึงภาษา C ที่เป็นคำสั่งของคอมไพเลอร์ชนิดต่าง ๆ ได้แก่ Microsoft C, Quick C และ Turbo C รวมถึง Turbo C++ โดยการกล่าวอ้างอิงถึง ANSI C เป็นพื้นฐานที่สำคัญ ดังนั้นหนังสือเล่มนี้นับว่าเป็นการเขียนโปรแกรมภาษา C อย่างแท้จริง ทำให้ผู้ที่ศึกษาค้นคว้าทางด้านภาษา C สามารถเข้าใจในภาษา C อย่างลึกซึ้งทั้งในมุมกว้างและในทางเจาะลึก และมีความมั่นใจในการเขียนโปรแกรมภาษา C อย่างมาก ซึ่งสามารถนำไปสู่การเป็นโปรแกรมเมอร์ทางภาษา C อย่างเต็มตัว และยังได้รวบรวมคำสั่งชนิดต่าง ๆ ของภาษา C รวมทั้งการเรียกใช้อินเทอร์รัพต์ และข้อมูลที่สำคัญในการเขียนโปรแกรม โดยสรุปไว้เป็นภาคผนวก ทำให้สะดวกต่อการค้นหาข้อมูลเพื่อใช้ในการเขียนโปรแกรม



- โปรแกรมกราฟิกและเมาส์
- ข้อมูลสำหรับการเขียนโปรแกรม
- รวมคำสั่งพร้อมโปรแกรมตัวอย่าง เพื่อการค้นหาที่รวดเร็ว
- คำสั่งโปรแกรมครอบคลุมถึง Microsoft C, และ Turbo C++